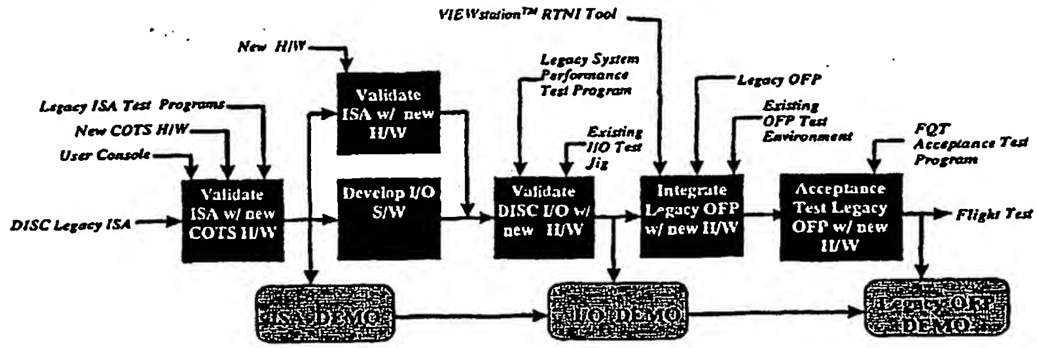


Fig. 5. First Step – Baseline New Hardware with Legacy OFP

functionality into an existing platform. Tailoring the RePLACE™ DISC Legacy ISA is required to match the unique features and characteristics of the legacy machine; tailoring the VIEWstation™ is required to integrate with the legacy software toolset. As Figure 5 illustrates, the steps in achieving validation include validating the execution of the ISA within the new microprocessor, validating the I/O characteristics using the new microprocessor, and integrating and running acceptance tests of the legacy OFP with the new hardware.

SUMMARY

The RePLACE™ technology offers significant supportability, producibility and performance improvements through the introduction of COTS-based state-of-the-art hardware; it maintains backward compatibility with existing OFPs while providing the means to affordably migrate to state-of-the-art hardware and software technology. This results in significant savings, offers substantial risk reduction, and promotes incremental upgrade opportunities for future enhancements. Also, RePLACE™ requires no custom hardware — it is an embedded software technology.

The RePLACE™ engine can be supplied with a turnkey box or with a set of open system COTS board level products. Also, PC-based configuration, control and monitoring software (VIEWstation™) is available to provide setup, maintenance, and user interaction with the RePLACE™ engine. Although RePLACE™ was conceived for on-board avionics computer replacement strategies, it is equally effective to other embedded computer applications, such as automated test equipment, trainers, and integrated support environments.

Under AFRL's Reconfigurable Avionics Computer Emulator (RACE) project, contract number F04606-95-D-0017, TRW is developing a COTS microprocessor demonstration of the F-16 General Avionics Computer (GAC) using RePLACE™ technology. TRW is also developing COTS replacement boxes for several other military aircraft targets with a goal of flight testing this technology in the near future.

Candidates Sought for Election to AESS Board of Governors

Eight AESS members will be elected to three-year terms (2000-2002) as members of the AESS Board of Governors (BoG) at the next meeting of the Board on April 21. The Nominating Committee is responsible for developing a slate of nominees for this election with the broadest possible representation of interests of our society membership.

I encourage members to consider this opportunity to suggest candidates with strong professional credentials and dedicated interest in the Society's success. All suggestions will be seriously considered.

The requirements for nomination, besides AESS membership, are the capability and resources to attend two of three BoG meetings per year and to devote several additional hours per month to AESS affairs.

Please send nominations for candidates by 30 March 1999 to Charles Gager, Nominating Committee Chair, or to any officer or BoG member. Gager can be reached by E-mail at: (c.gager@ieee.org); addresses and phone numbers for Gager and other AESS officials are given inside the back cover of this magazine.

Replacement Strategy for Aging Avionics Computers

Jahn Luke
Air Force Research Laboratory (AFRL)
and
John W. Bittorie, William J. Cannon, Douglas G. Haldeman,
TRW Avionics System Division / Avionics Engineering Centers

ABSTRACT

With decreasing defense dollars available to purchase new military aircraft, the inventory of existing aircraft will have to last many more years than originally anticipated. As the avionics computers on these aging aircraft get older, they become more expensive to maintain due to parts obsolescence. In addition, expanding missions and changing requirements lead to growth in the embedded software which, in turn, requires additional processing and memory capacity. Both factors, parts obsolescence and new processing capacity, result in the need to replace the old computer hardware with newer, more capable microprocessor technology. New microprocessors, however, are not compatible with the older computer instruction set architectures. This generally requires the embedded software in these computers to be rewritten. A significant savings — estimated in the billions of dollars — could be achieved in these upgrades if the new computers could execute the old embedded code along with any new code to be added.

This paper describes a commercial-off-the-shelf (COTS)-based form, fit, function, and interface (F^3I) replacement strategy for legacy avionics computers that can reuse existing avionics code "as is" while providing a flexible framework for incremental upgrades and managed change. It is based on a real-time embedded software technology that executes legacy binary code on the latest generation COTS microprocessors. This technology, developed by TRW and being applied under the sponsorship of the Air Force Research Lab (AFRL), promises performance improvements of 5-10 times that of the legacy avionics computer that it replaces. It also promises a 4X decrease in cost and schedule over rewriting the code and provides a "known good" starting point for incremental upgrades of the embedded flight software. Code revalidation cost and risk are minimized since the structure of the embedded code is not changed, allowing the replacement computer to be retested at the "blackbox" level using existing qualification tests.

DESCRIPTION

TRW has developed a generic COTS-based software technology that allows a user to: 1) replace their obsolete computers with commercial COTS-based hardware; 2) continue to use their existing Operational Flight Software (OFP) without modification; and

Author's Current Address:
J. Luke, Air Force Research Laboratory (AFRL), Wright-Patterson AFB, OH 45433, USA; J.W. Bittorie, W.J. Cannon and D.G. Haldeman, Military address unknown.

Based on a presentation at NAECON '98.
0885/8985/99/$10.00 © 1999 IEEE

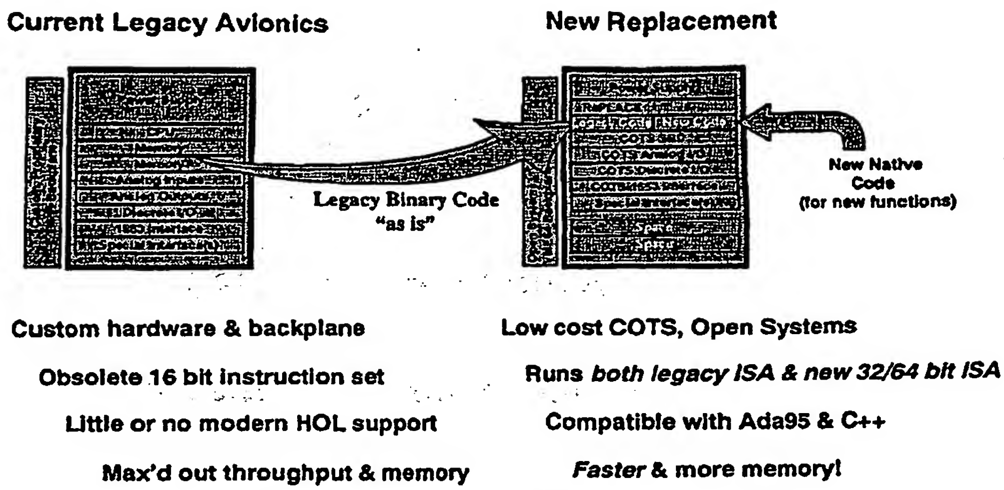
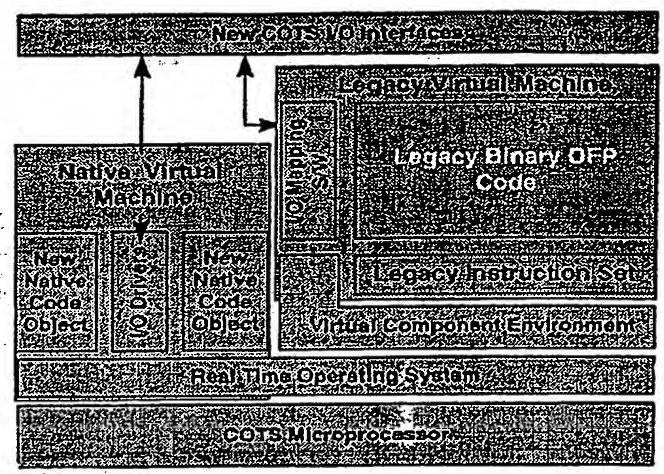


Fig. 1. RePLACE™ – The Technology

3) incrementally upgrade their hardware and software to support new and modified capabilities. We refer to this technology as RePLACE™ — Reconfigurable Processor for Legacy Avionics Code Execution. It allows a user to capitalize on technology advances by replacement of their obsolete and outdated hardware with militarized COTS equipment and open systems standards without incurring the associated cost of rewriting the legacy software to execute on the new hardware. RePLACE™ allows DoD dollars to be spent on solving supportability and performance problems and adding the new capabilities needed, rather than recapturing current capabilities in the new hardware.

Figure 1 depicts the use of RePLACE™ in migrating from a current legacy avionics line replaceable unit (LRU) to a new COTS-based replacement (CRB) box. Typically, today's legacy avionics are based on custom hardware and proprietary backplanes with an obsolete 16-bit instruction set. Little or no *modern* higher order language (HOL) support is available to support software for this equipment. In addition, legacy avionics are often limited in terms of throughput and memory. A COTS-based, open system replacement strategy with RePLACE™ provides a low cost of entry strategy for COTS microprocessor technology insertion. RePLACE™ allows the execution of both the legacy instruction set architecture (ISA) as well as the new native ISA. Native code execution is supported by advanced HOLs such as Ada95 and C++. In addition, legacy code can be executed much faster in the new hardware and more memory is available for needed upgrades.

A software perspective of RePLACE™ is illustrated in Figure 2. Sitting on top of the COTS microprocessor and POSIX real-time operation system (RTOS) are two virtual machines – the legacy virtual machine and the native virtual machine. Within the legacy virtual machine space are four key software components: the legacy



[redacted] The Software View of RePLACE™

instruction set engine; the legacy OFP code binary image; the input/output (I/O) mapping software; and the virtual component environment (VCE). The legacy instruction set engine contains unique cache-optimized code developed by TRW to execute the legacy binary OFP code. The I/O mapping software maps the new COTS interface devices to the legacy interface devices. The VCE allows for the efficient transition between the native and legacy virtual environments — allowing the transfer of data and concurrent execution of both legacy and new native code.

The key features of RePLACE™ may be summarized by the following points:

- RePLACE™ relies on the use of state-of-the-art COTS microprocessors and open system standards that improve both the legacy system's producibility and supportability.

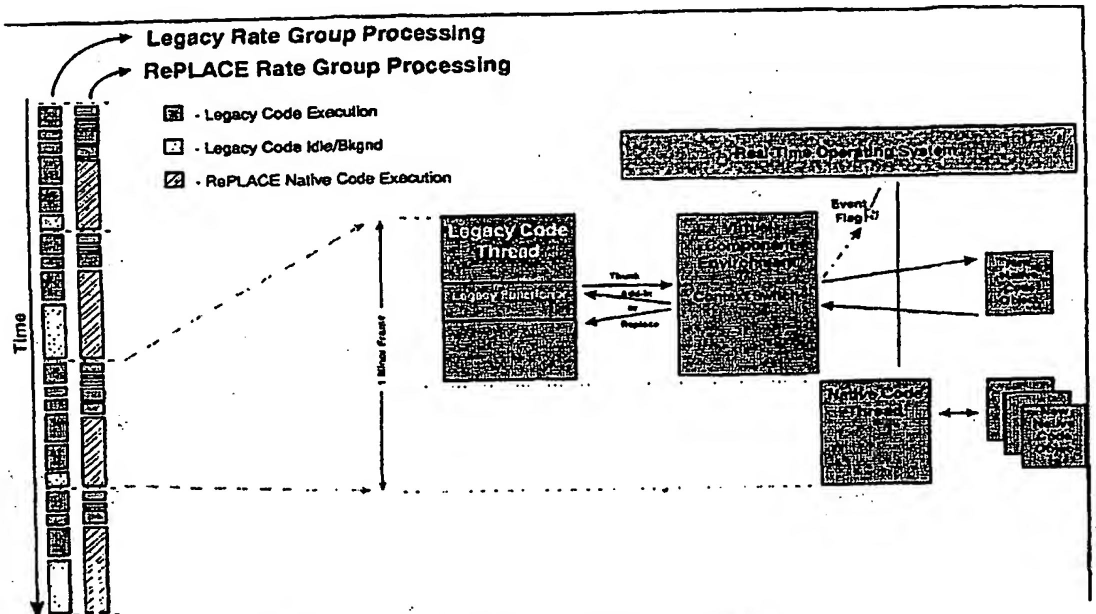


Fig. 3. Concurrent Legacy and Native Code Execution

- RePLACE™'s unique cache-optimized approach runs legacy code from 2 to 10 times faster than the original legacy system. This provides extra compute power for new functions. In addition, the performance of the legacy virtual machine is linearly scaleable with the performance of the COTS microprocessor technology that is being used. That is, as the internal clock speed of the microprocessor chip goes up, the legacy instruction execution speed increases linearly.
- The instruction set engine in RePLACE™ is adaptable to diverse instruction set architectures, including, for example, MIL-STD-1750A, Z-8002, and AN/AYK-14A. This makes possible the replacement of multiple diverse legacy avionics LRUs with a common set of avionics hardware modules on a given platform or even across different platforms.
- RePLACE™ provides input/output (I/O) mapping software that exactly matches the new I/O COTS devices to the legacy I/O devices — providing a "drop-in" environment for the unmodified legacy OFP with little or no knowledge of the original code.
- Both legacy and new native software execute concurrently in separate virtual spaces. This promotes incremental addition of new capabilities and gradual transition to new code.
- Instruction execution speed matching can be initiated in critical sections of the OFP to provide legacy OFP timing adjustments for timing sensitive code.
- Real-time non-intrusive (RTNI) monitoring of legacy software is now practical and is built into the replacement avionics – dramatically enhancing the observability and testability of the embedded legacy software.

In order to illustrate the concurrent execution of legacy and native code in a RePLACE™ dual instruction set computer (DISC) environment, Figure 3 depicts the concept for the case in which the legacy code remains in control and new software enhancements are introduced. On the far left are two time lines showing the various rate group processing tasks running in the legacy machine and in the COTS replacement box (CRB). In the case of the CRB, the original rate groups are executed in a much shorter time frame within any given minor cycle. This leaves additional processor throughput at the end of each minor cycle to add new software running in the native ISA. Through the virtual component environment (VCE) context switch mechanism, new native code can be introduced to replace existing legacy code or add to the existing legacy code. Alternately, event flags can be set to augment the legacy code thread as illustrated. Note that legacy instruction execution speed matching can be introduced for timing-sensitive code. Also, the technology includes the capability to disable legacy code outputs without legacy OFP cognizance, providing a convenient mechanism to switch off functions in the legacy code without being

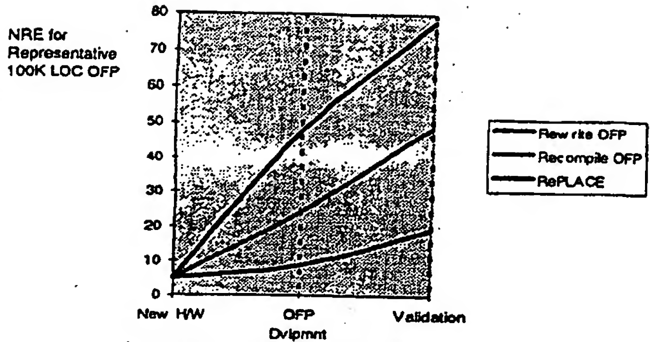


Fig. 4. Representative NRE Cost Tradeoff (Man-Years)

intimately familiar with the legacy code structure. In all cases, the legacy binary OFP code remains intact.

Another key component of the RePLACE™ strategy is the availability of a source level, symbolic user console for system developers. A tool is being developed to facilitate the use of RePLACE™ in modern microprocessor technologies and is referred to as the Virtual Integration Environment Workstation, or VIEWstation™. It is tightly integrated with the COTS native code Integrated Development Environment (IDE) and commercial tools that are selected to support new native code development. It is loosely integrated with the legacy code tools. It provides a source level, symbolic configurator tool to assist the software developers in mixing and matching legacy and native code; it also provides in-target debugging and real-time non-intrusive (RTNI) monitoring of the legacy code. VIEWstation™ incorporates COTS graphical data analysis tools and an interactive symbol browser/editor. Examples of the use of VIEWstation™ include downloading and disassembling software binaries, displaying real-time debug and monitoring data, performing and displaying timing characteristics, creating and deleting events, and developing "add-in" code that is executed in response to specific user-specified events in the legacy code.

RESULTS

RePLACE™ has been demonstrated using an F-16 HUD binary OFP executing in real-time on a Power PC 603e microprocessor. The legacy HUD is a 1750A machine that executes at approximately 1 MIP. The unmodified HUD OFP binary code was successfully demonstrated executing on a Power PC at a speed of more than 10 "1750A" MIPS – within a normal real-time F-16 operational environment. The addition of new native code, written in C++, was also demonstrated to augment the real-time HUD display. As part of this demonstration, the attributes and benefits of VIEWstation™ were illustrated, including the non-intrusive monitoring of key HUD OFP parameters during real-time operation.

Under contract to AFRL, the MIL-STD-1750A dual instruction set computer (DISC) is currently undergoing a complete validation using the official acceptance test procedures and verification software originally developed and supported by the Systems Engineering Avionics Facility (SEAFAC).

BENEFITS

The use of RePLACE™ technology assumes the need to replace legacy hardware with newer, more capable and more supportable microprocessor technology. The objective of RePLACE™ is to allow for hardware upgrades without incurring the associated cost to adapt the software to the new hardware – typically a significant percentage of an upgrade cost. A comparison of various alternatives to software adaptation are shown in Figure 4 for the cases of rewriting the OFP, rehosting the OFP using a retargeted compiler, or using RePLACE™. The non-recurring costs for OFP development are, as would be expected, much higher for the case of an OFP rewrite; to a lesser extent but still significant are the costs associated with an OFP rehost. This is because a rehost activity must still address new machine dependencies and the immaturity of the associated software tools that are targeted for the new hardware. In addition, under the OFP rehost strategy, incremental software upgrades are difficult to implement. As the bottom curve illustrates, the cost for OFP development with a RePLACE™ approach is extremely small – because the existing binary OFP code is used as is – unmodified. The costs included in the figure are representative of the time to tailor the I/O mapping software to support the new hardware and to incorporate legacy software tools into VIEWstation™.

Code revalidation costs are significant for all three approaches. However, these costs are minimized with RePLACE™ since the structure of the embedded code is not changed, allowing the replacement computer to be retested at the "blackbox" level using the existing acceptance test plans and procedures.

TRW has developed a set of analytical tools to support this type of tradeoff for different user scenarios.

UPGRADE STRATEGIES WITH REPLACE™

Potential upgrade strategies using RePLACE™ cover a wide spectrum of upgrade possibilities. These include the introduction of a new ruggedized COTS replacement box for an existing legacy avionics LRU. It also includes the introduction of a new microprocessor replacement module for insertion into an existing avionics LRU. Finally, it includes the use of RePLACE™ as a tool to mitigate the inherent risks associated with introducing both new hardware and software into a platform at the same time. Figure 5 illustrates the use of RePLACE™ for baselining the new COTS-based hardware with the legacy OFP as a first step in the process of adding new